US-PAT-NO: 5457476

DOCUMENT-IDENTIFIER: US 5457476 A

TITLE: Method for controlling a computerized organizer

---------- KWIC ---------

In FIG. 9, the view has been changed to the **to-do list** function for the single day as originally shown in FIG. 3b. Note entry 86 remains visible and to-do entries 96, 98, and 100 are now present. Each of the to-do entries has a **check** box 102, a priority box 104, and duration bar 106. **Check** box 102 allows entries to be designated as completed by simply entering a **check** mark in the box. Unmarked entries are carried forward to the next day so that undone tasks automatically remain on the **to-do list**. Priority box 104 displays a pull-down menu of choices to designate the priority of each to-do entry and is further described with reference to FIG. 10. Duration bar 106 behaves similarly to an appointment's duration bar, so that notes can be attached to to-do entries in the same manner as described previously with reference to the schedule function. The start and stop fields hold the difference in the two forms of the edit window. Since a to-do entry does not have a start or stop time, these fields in the edit window are not alterable. New entries can be added to the **to-do list** by making a breaker bar 108 with the stylus 38. A breaker bar is made by moving the stylus 38 horizontally across the width of screen 42. Breaker bar formation is decribed in detail in co-pending U.S. patent application Ser. No. 07/868,013 filed Apr. 13, 1992, on behalf of Tchao et al, entitled "Method for Manipulating Notes on a Computer Display" now U.S. Pat. No. 5,398,310, and assigned to the assignee of the instant application, the disclosure of which is hereby incorporated by reference in its entirety. A new **check** box 110, priority box 112, and duration bar 114 appear as the stylus 38 is removed to display the addition of a new to-do entry 116.

Similarly, entries can also be easily rearranged by moving entries from the note area 74 to a desired **to-do list,** as illustrated in FIG. 13. As a to-do entry, note entry 86 now has an associated priority box 118, **check** box 120, and duration bar 122. Although not illustrated, to-do entries can just as easily become note entries by the reverse procedure of dragging an entry from a **to-do list** into note area 74. The capability of moving data and information as described exemplifies the convenience and efficiency that a digital assistant incorporating the methods of the present invention can provide.

FIG. 14 is a flow diagram of a process in accordance with the present invention for performing the functions illustrated in FIGS. 3a, 3b, and 4-13. The process begins at 124, and in a step 126, the schedule/to-do list function is activated. This activation occurs with the selection of the control button 52' as previously described and results in the production of the default schedule function view. Once activated, the next step 128 **checks** for the selection of the calendar/schedule view by the selection of the calendar button 129 (FIG. 3a). If the schedule function is selected, the process continues in step 130 with the processing of schedule functions, which are further described with

reference to FIGS. 15-20. If the schedule function has not been selected, step 132 **checks** for the selection of the **to-do list** function. If selection is detected, step 134 processes **to-do list** functions, further described in FIGS. 21-24. Upon completion of steps 130 or 134, or with a negative result in decision step 132, the process returns to step 128 to repeat the loop.

US-PAT-NO: 6685478

DOCUMENT-IDENTIFIER: US 6685478 B2

TITLE: Inexpensive computer-aided learning methods and apparatus for learners

---------- KWIC ---------

Claims Text - CLTX (56):

56. A computerized method for managing learning activities, the method comprising: **tracking** a learning activity of a student, becoming aware of the student's understanding in an area; creating a report corresponding to at least some of the **tracked** learning activity; receiving a request from an enterprise to access the report; determining whether the enterprise could access the report; responsive to determining that the enterprise could access the report, retrieving at least a portion of the report; transmitting the retrieved at least a portion of the report to the enterprise; and determining whether a student should be involved in a learning activity based on a task the student is to perform, wherein the enterprise is charged, wherein based on the charging, a **student is allowed to be involved in a learning activity, and the enterprise is allowed to access the report,** and wherein the method is implemented in a network environment.

Claims Text - CLTX (78):

78. A computer readable media containing computer program code that is useful for managing learning activities, said code when executed by a computer, causing the computer to perform a method comprising: **tracking** a learning activity of a student, becoming aware of the student's understanding in an area; creating a report corresponding to at least some of the **tracked** learning activity; receiving a request from an enterprise to access the report; determining whether the enterprise could access the report; responsive to determining that the enterprise could access the report, retrieving at least a portion of the report; transmitting the retrieved at least a portion of the report to the enterprise; and determining whether a student should be involved in a learning activity based on a task the student is to perform, wherein the enterprise is charged, wherein based on the charging, a **student is allowed to be involved in a learning activity, and the enterprise is allowed to access the report,** and wherein the method is implemented in a network environment.

Current US Original Classification - CCOR (1):

434/219

Current US Cross Reference Classification - CCXR (1):

434/118

Current US Cross Reference Classification - CCXR (2):
434/350

US-PAT-NO: 5826252

DOCUMENT-IDENTIFIER: US 5826252 A

TITLE: System for managing multiple projects of similar type using dynamically updated global database

---------- KWIC ---------

In this example of a project management program, the system carries out its project tracking functions essentially by list management. A basic sequence for management of a typical list of tasks representing any task component of the overall project is illustrated in FIG. 3. As the project task component is performed, the tasks on the list are **checked** by the project manager and members of the project team. If a task is performed, it is so noted on the list. If a task is modified locally, approval is obtained from the customer and the vendor supervisory group and the modified **task is entered on the list**. If a locally modified **task is superceded, it is deleted from the list and a new task** item may be entered. Checklist items that originate from a Master Template from the Global Database are not permitted to be modified or deleted from the list. If a master template item is to be superceded, the item is marked "not applicable", then a new item can be added at the discretion of the project manager. At the completion of the project, new items are reviewed by the vendor supervisory group.

US-PAT-NO: 6065012

DOCUMENT-IDENTIFIER: US 6065012 A

TITLE: System and method for displaying and manipulating user-relevant data

---------- KWIC ---------

In another implementation, the code within an appropriate data binding HTML table is capable of being executed to directly instruct the appropriate control module to manipulate the underlying data. For example, each row in the task data binding HTML table 470 represents a **check** box and some text that describes a particular **task as shown in the list** of tasks 215c within the view 210. The state of the **check** box represents the completeness of the particular task. When the user desires to **check** off one of the tasks 215c as being complete without invoking the program module (a task module 415), the user positions the cursor over one of the tasks 215c and clicks a button on the mouse 42. In response, code within a row of the task data binding HTML table 470 executes to send a notification to the task control module 435 with row and column information to identify which tasks was modified. The modification to the task data items 315 is then performed by the task control module 435 in order to permanently save the modification. This is advantageously performed without the time consuming need to open the task module 415 and set the status of the particular task to be complete within the task data items 315.

US-PAT-NO: 6044387

DOCUMENT-IDENTIFIER: US 6044387 A

TITLE: Single command editing of multiple files

---------- KWIC ---------

When initiating the spell **check** editing operation, the user is also provided with an option in a **check** box 110 to add the task of reviewing pages that are appear to include one or more misspellings to the ToDo List. Choosing this option will cause the **task of reviewing the list** of documents that apparently include spelling errors to be inserted as an entry in the ToDo List so that the spelling errors can be reviewed by the user at a later time. Since a web site may include hundreds of such documents that must be spell **checked**, it may be useful to thus implement the spell **checking** process as a background **task so that the list** of pages or documents that apparently include spelling errors is generated overnight or during times when the client computer is not being used for other persons, so that the apparent spelling errors in the documents included in the list can be reviewed by the user at a later time.

Also included in the **Check** Spelling dialog box of FIG. 4 are a Stop button 124, an Edit Page button 126, which is usable once the list of documents with spelling errors has been completed, and an Add Task button 128 that enables the user to assign the review of spelling errors in one or more documents to a specific person by adding that review as a **task on the ToDo List** for implementation at a later time. In addition, a Help button 130, and a Close button 132 are included.

The user can select any one of the documents listed in window 82 for review of the apparently misspelled words, causing the document to be opened at the first misspelled word, by selecting Edit Page button 86. Alternatively, the user can select one of more of the documents and add the task of reviewing the one or more documents and the apparently misspelled words found thereon to the ToDo **List by selecting Add Task** button 88. When a user edits the page that is included within the list of documents in which apparently misspelled words were found, each of the apparently misspelled words will be highlighted within the text at the point it appears in the document and the user will be given the opportunity to change the spelling of the word or to ignore it. Since some words will appear in documents that are not found in a spell **check** dictionary provided with the program or in a user dictionary generated by adding words that are not in the general dictionary, a word identified as misspelled may not be. If the user elects to add a word to the user dictionary, that word will be included with the words presumed to be correctly spelled in any future spell **check** editing operation. Alternatively, if the user elects to ignore the apparent misspelling of the word, no further action by the user is required and the next apparently misspelled word, if any, will be highlighted in the document.

After the find and replace operation is initiated, the program opens each of

the documents that the user selected to be searched for the text that was specified and produces a list much like that shown in FIG. 1 in which each document containing the characters specified by the user is found. Again, the status column includes a symbol indicating whether the documents in the list have been reviewed or added to a ToDo List as a task. In this embodiment, the symbol in the status column is colored red until the document adjacent the symbol has been either reviewed, or a decision has been made to defer the editing by adding the document as a task to a ToDo List. Once the document is reviewed, the status symbol changes from red to yellow and the word "Edited" appears next to the symbol. Similarly, if the review of the document is deferred by placing it on a ToDo List, the symbol changes to a yellow color and the text "Added Task" appears next to the symbol. These same changes in the indicated status of a document described above also occur in connection with a spell check editing operation, i.e., once a user has reviewed the indicated errors in a document or deferred the review by creating a task on the ToDo List, the status symbol changes color and the appropriate text is applied in the status column.

A block 198 begins opening the files in the list that was built in either block 192 or 194, beginning with file n. Initially, for n=0, the first file in the list is open and in a block 200, the program initiates the editing operation on file n. In a block 202, the program determines whether the editing operation selected by the user is applicable to file n. For example, if the user has elected to spell check the files in the list that was built, each word occurring in the current file that has been opened will be checked by comparing it against words in a dictionary to determine if any word can be identified that is apparently misspelled. Alternatively, and in a similar fashion, a find and replace editing operation would be implemented on the currently opened file to determine if any text matching the characters specified by the user is found in the file. If so, the logic proceeds to a decision block 204, which determines if the user has chosen to display the summary review or list of the tables to which the editing operation is applicable. It should be noted that if the user has elected to spell check the files and further, has chosen to defer the review of the files in which apparently misspelled words were found, the summary review list of files will not be shown. Instead, any currently opened file that is found applicable for the editing operation would be added to a ToDo List as an additional task, as noted in a block 212, and the user will be presented with a count of the number of files added to the ToDo List. While files are being checked for misspelled words in this mode, a status bar is displayed to show the progress. This option to defer review of the files is only available when doing spell checking in the embodiment included in FRONTPAGE 97. If the user is conducting a find and replace editing operation or has chosen not to defer processing of the files, an affirmative response to decision block 204 leads to a decision block 206. This decision block determines whether the summary review list of files is showing, and if so, a block 208 provides for adding the current file as an additional record in the summary list. A negative response to decision block 206 causes the summary review list to be displayed, and thereafter, the logic proceeds to block 208 to add the record to the summary review list. Following block 208 or block 212, a block 214 increments the counter n, and the logic proceeds to a decision block 216. In decision block 216, the program determines if the number of files in the list that was built in block 192 or block 194 is greater than n. If so, not all of the files in this list have been processed, and the logic proceeds to block 198 to open the next file in the list of files built in block 192 or in

block 194.

However, if the user has not closed the process in decision block 222, a decision block 230 determines if the user has chosen to add the review of one or more files selected from the list as a task on the ToDo List. If so, a block 232 carries out this operation. However, if the user has instead chosen to review a file, the page or file is opened in a block 234. A block 236 provides for performing reviewing the results of the editing operation in the open document. In the case of spell checking, the user would be given the option of changing the spelling of an apparently misspelled word, skipping to the next apparently misspelled word, or ignoring the misspelled word. Once all of the misspelled words indicated in a file have been processed, the file or page is removed from the list of documents yet to be processed. Actually, this step is implemented simply by changing the status indicator of the file in the summary review list.

US-PAT-NO: 5835758

DOCUMENT-IDENTIFIER: US 5835758 A

TITLE: Method and system for respresenting and processing physical and conceptual entities

--------- KWIC ---------

Abstract Text - ABTX (1):

A computer-based method and system is described for representing, storing, analyzing, processing, and communicating conceptual and physical entities. On a computer display screen, means are presented for users to create a palette of elements to define and describe entities and matters of interest to them. With a palette that they create, or with a previously defined palette, users create computer-based maps containing specific instances, called items, of one or more elements in the palette. Users can link items with line segments, arrows, or other connectors to show the relationships between them. Data, visual, and other attributes may also be defined for the links between items, the maps on which the items and links are defined, and case folders which index the maps and other associated computer-based documents and objects. Associations and connections can be established between a map, or the items or links on it, and other maps and other computer-based documents or objects such as spreadsheets, word processor files, graphical objects, audio objects, and video objects. Rules, heuristics, and norms may be specified for the palette elements, based on which maps may be parsed to see if they are correct, consistent, and complete with regard to the items they contain and the relationships indicated by the links between them. Queries can be made and **reports** can be generated based on the data and other attributes defined for the items, links, maps, and cases. Items, links, maps, cases, and other computer-based documents and objects can be shared among various users on a computer network. Details regarding actions or responses relating to items, links, maps, cases, and other computer-based documents and objects can be maintained in computer-based organizers, and computer messages can be generated to remind or alert users about their status and the actions to be taken relating to them. A particular embodiment of the present invention that is described here in detail pertains to the domain of purposeful human activities, inter alia goal-setting, problem-solving, decision-making, planning, and action-implementation in individual, group, and organizational settings.

Brief Summary Text - BSTX (9):

Flowcharting and other diagramming tools have visual elements, i.e. shapes and lines, that can show connections between various elements. However the shapes do not carry labels or other descriptors, nor do the tools offer any syntax that can create a consistent visual language. Further, they do not allow specific attributes to be defined for various shapes. As a result, the visual elements cannot be analyzed or queried, and useful **reports** cannot be generated.

Brief Summary Text - BSTX (15):

Semantic hypertext systems have very limited analysis or **reporting** capability because they do not capture important data attributes relating to the items they represent. E-mail products and networked databases do not have a structured language or frameworks for clarifying issues or showing how they are interconnected.

Brief Summary Text - BSTX (20):

Rules, heuristics, and norms may be specified in the present invention for the palette elements individually and for assemblages of them collectively. Based on such rules, heuristics, and norms, maps may be parsed to see if they are correct, consistent, and complete with regard to the items they contain and the relationships indicated by the links between them. Queries can be made and **reports** can be generated based on the data and other attributes defined for the items, links, maps, and cases. Items, links, maps, cases, and other computer-based documents and objects can be shared among various users on a computer network. Users can send messages to each other over the network, along with attachments of or references to items, links, maps, cases, and other computer-based documents and objects of interest.

Detailed Description Text - DETX (6):

In **report** definition module 13, users specify the formats 131 for various **reports** that they want to create, based on the attributes defined earlier for items, maps, links, cases, and text documents. This module also creates the query dialog boxes 133 users will need to define queries, and dialog boxes for selection and sorting 135 data for generating various standard and customizable **reports**.

Detailed Description Text - DETX (8):

Analysis definition module 15 allows users to define various rules, conditions, and heuristics that will be used to check whether attributes for items on a map and the links between those items are correctly specified, whether attributes of linked items are consistent with regard to user-defined criteria, and whether maps are complete in terms of the items that are present, or not present, on them. This module generates sub-routines 152 to analyze maps with regard to the items and links in them for correctness, consistency, and completeness, and **report** formats 154 for the results of such analyses.

Detailed Description Text - DETX (13):

The particular embodiment of the invention described herein consists of the following modules, as shown in FIG. 2: a visual interface manager 20, a database manager 25, a database 28, a **report** module 202, an action module 204, a messaging module 206, an analyzer module 208, an access module 210, a text module 212, a case module 214, a map module 216, and a help module 218.

Detailed Description Text - DETX (25):

Users can generate **reports** based on their requirements. When a user selects

the command to generate a report, visual interface manager 20 sends a message to report module 202 to display a dialog to select details of the report such as--the sources of information to be included in the report and the fields and records to be queried. Report module 202 then defines the report, displays it to the user, and allows the user to print or save the report. When the user selects a command to load a saved report, visual interface manager 20 passes the command to report module 202 which displays all reports which have been saved. After the user selects the report module 202 to be loaded, the report loads it from secondary storage, extracts the required information, and displays it to the user.

Detailed Description Text - DETX (26):

Map analyzer module 208 is used to analyze maps for completeness, correctness, and consistency with regard to the items they contain and the links between the items based on various rules, conditions, and heuristics. Map view manager 32 retrieves information about the items and links between them from map document 37 and passes this information to map analyzer module 208. The map analyzer module 208 sends a message to database manager 25 via map view manager 32 and visual interface manager 20 with details of the map to be analyzed. Database manager 25 parses the map using the parent-child relationships between its items and the connections between them. The parsing information is passed to map analyzer module 208 which processes it on the basis of normative rules, conditions, and heuristics with regard to the item attributes, the presence or absence of other items, and the links between items. Map analyzer module 208 passes the results of the analysis to map view manager 32 for display on the screen. The user can also print the report via report module 202.

Detailed Description Text - DETX (31):

The user can create a to-do list which may be free-standing or attached to actionable items on a map or other document. This is accomplished via action module 204 which handles the tasks of creating and saving to-do items, and transferring relevant to-do entries from existing items or memos via database manager 25. The user can invoke the to-do list via visual interface manager 20 which passes control to action module 204. Various parameters to generate alerts and reminders--such as dates, times and monitoring frequencies--can also be specified in action module 204. These parameters are checked against item attributes, dates attached to to-do entries, current time and date, and other variable to trigger reminders and other such messages to alert users regarding the status of various items and documents. These reminders and alerts are displayed via visual interface manager 20 and accompanied, if so specified, with audible and other signals.

Detailed Description Text - DETX (33):

The advantages and benefits of the particular embodiment of the invention, in the domain of purposeful human activities, described here include: a) It provides a concise, flexible, and simple visual and verbal method and system for precise thinking, communication, implementation, and follow-up action; b) It creates an integrated environment for dealing with all key issues that are relevant in the context of its use; c) It provides means for delegating, tracking, monitoring, and updating of issues, actions, and results over a

distributed computer network; d) It delivers in a just-in-time fashion tools, guidance, and process know-how to improve goal-setting, problem-solving, decision-making, planning, analysis, and other such activities; e) It promotes improved comprehension and clarity about complex situations and issues; f) It fosters a sense of context and connectivity by showing how details fit in the "big picture" perspective; g) It captures data attributes that enable analysis, report generation, and other kinds of processing; h) It provides feedback to users about the correctness and completeness of the documents they create based on an internal system of syntax, semantics, and data attributes. As a result of all the above features, advantages, and benefits, the present invention enhances clarity, communication, coordination, and collaboration among its users, thereby leading to greater productivity, effectiveness, and quality in their efforts and results.

Detailed Description Text - DETX (47):

Tasks, Plans, and other items can be delegated or assigned by one user to others for various kinds of actions such as--Comment, Do, Annotate, Discuss, Reply, etc. Deadlines for such action can be specified. Once delegated or assigned over the network, they can be tagged and monitored to see if specified actions have been taken by the times defined. If no action has been taken and reported by the recipient, the system can put out an alarm to the dispatcher to alert him/her that a specific delegated item has not been acted on or responded to. If the recipient completes the action, he/she can mark off the item as Done, or attach other status notes, and send it back to the original dispatcher. A Status Update will then inform the sender of the status of the work that was originally sent out. Such tracking and monitoring and the triggering of alerts, reminders, or updates shall be an optional feature--the option to have them on or off shall be decided by users, depending on the organizational norms for such practices. At one extreme, objects and documents can be fully passive, i.e. no tagging, tracking, or updating will be possible. At the other extreme, automatic processes can be initiated to tag and track, remind or alert senders and receivers about status and other relevant variables, and provide status update reports periodically or on an "as needed" basis. These operations are managed by action module 204 shown in FIG. 2.

Detailed Description Text - DETX (49):

The present invention will connect with the users' day to day world by providing an internal to-do list manager and by interfacing with their external calendars, schedulers, to-do lists and other such planning, organizer, and action tools. The internal to-do list feature allows them to specify action items along with related start and end dates, priority, and status data. Such entries may be free-form, i.e. unattached to any item or map or other such entity of the present invention, or they could be attached to items, maps, memos or other such objects. Relevant data from memos, items, and other objects can be easily transferred to the to-do list to enable their implementation. A `reminder` facility allows users to attach remind dates and times to to-do items, memos, and other objects. The system date and time is checked against such remind dates and times to generate appropriate reminder messages. Such operations are managed by action module 204 shown in FIG. 2.

Detailed Description Text - DETX (50):

Reports can be generated by the present invention based on the attributes of items in maps and the attributes of links, maps, cases, and other documents as well. **Reporting** is accomplished by **report** module 202 (FIG. 5) which interfaces with database manager 25 to access data from database 28 to generate various **reports** 51.

Claims Text - CLTX (30):

8. The computer-implemented method of claim 5, further comprising generating **reports** based on said attributes of said instances of said elements.

US-PAT-NO: 6065012

DOCUMENT-IDENTIFIER: US 6065012 A

TITLE: System and method for displaying and manipulating user-relevant data

--------- KWIC ---------

Detailed Description Text - DETX (87):

In another implementation, the code within an appropriate data binding HTML table is capable of being executed to directly instruct the appropriate control module to manipulate the underlying data. For example, each row in the task data binding HTML table 470 represents a **check** box and some text that describes a particular **task as shown in the list** of tasks 215c within the view 210. The state of the **check** box represents the completeness of the particular task. When the user desires to **check** off one of the tasks 215c as being complete without invoking the program module (a task module 415), the user positions the cursor over one of the tasks 215c and clicks a button on the mouse 42. In response, code within a row of the task data binding HTML table 470 executes to send a notification to the task control module 435 with row and column information to identify which tasks was modified. The modification to the task data items 315 is then performed by the task control module 435 in order to permanently save the modification. This is advantageously performed without the time consuming need to open the task module 415 and set the status of the particular task to be complete within the task data items 315.

Current US Original Classification - CCOR (1):

707/102

Current US Cross Reference Classification - CCXR (2):

707/201